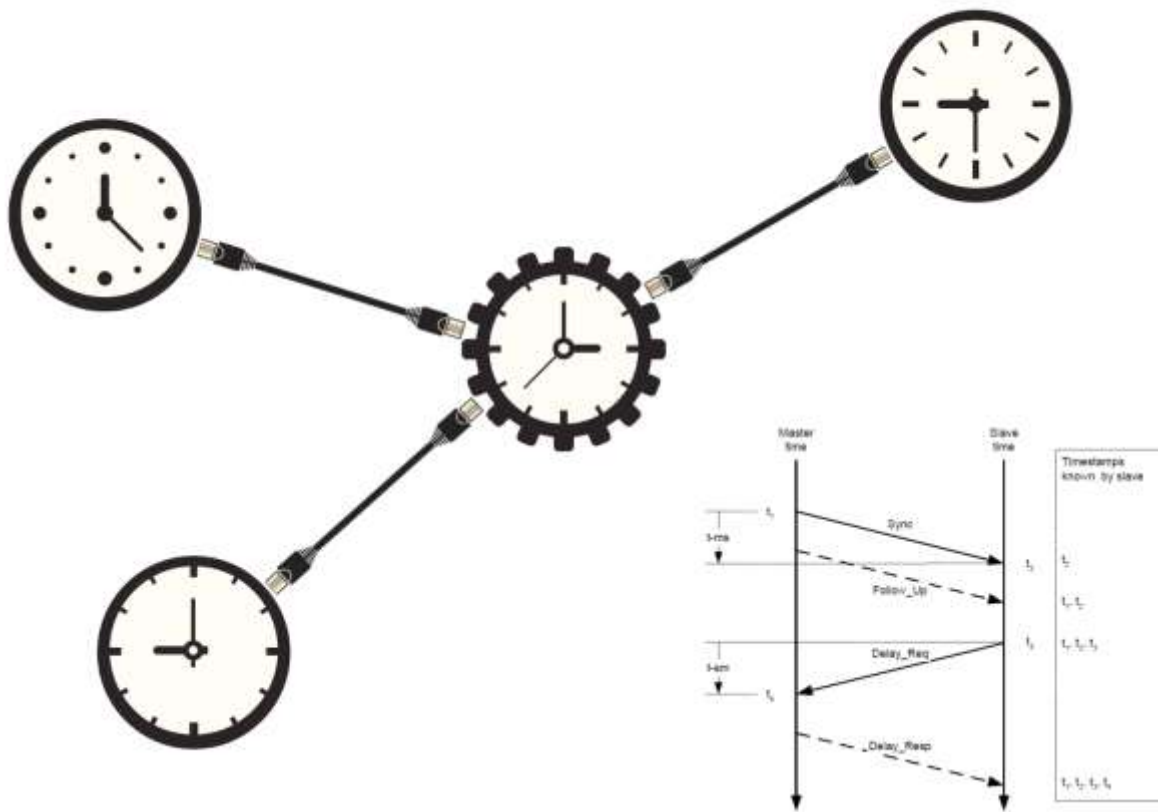


**TSEP**

Technical  
Software  
Engineering  
Plazotta

# IEEE 1588 PTP



## Der IEEE 1588 Standard

## Der IEEE 1588 Standard

Zeitsynchronisation zwischen mehreren Rechnern im Netzwerk war schon immer eine Herausforderung. Was sich im ersten Schritt als eine triviale Anforderung darstellt, zwei oder mehrere Rechner mit der identischen Zeit auszustatten, ist bei genauerer Betrachtung ein durchaus komplexes Problem. Die Problemstellung kann wie folgt definiert werden:

**Verteilung der Zeit einer sehr genauen Zeitquelle (Atomuhr, GPS) an einem oder mehreren Rechner im Netzwerk. Zur Verteilung der Zeit steht nur die vorhandene Netzwerkinfrastruktur zur Verfügung.**

Um dieses Problem zu lösen wurde bereits 1985 von David L. Mills an der Universität von Delaware das NTP Protokoll entwickelt und als RFC 958 veröffentlicht. Mit diesem Protokoll war es nun möglich mehrere Rechner an die Zeit eines Zeitservers anzugleichen. Die Genauigkeit die hierbei erreicht wird liegt im Mikrosekunde Bereich ( $1 \times 10^{-6}$  sec.) und war zur damaligen Zeit ausreichend.

Bereits 1990 wurde im T&M Bereich die Notwendigkeit erkannt über deutlich genauere Systeme zu verfügen. Zu dieser Zeit waren zwar bereits auch GPS System verfügbar, die eine deutlich bessere Genauigkeit bereitstellen konnten, jedoch war diese Alternative preislich nicht attraktiv. Zusätzlich sollte die Anforderungen an eine verbesserte Genauigkeit mit der vorhandenen Infrastruktur erreicht werden. So wurde Anfang der 90ziger Jahren bei Hewlett-Packard eine Arbeitsgruppe um John C. Eidson gegründet, die ein Konzept zur Verbesserung des NTP Standards ausarbeiten sollte. Die im Jahr 2000 von John C. Eidson veröffentlichten Ergebnisse waren für die T&M Firmen so interessant, das man einen eigenen IEEE Standard für diese neue Art von Zeitsynchronisation ins Leben rufen wollte. 2002 war es dann soweit, die IEEE 1588-2002 war fertiggestellt und freigegeben. Mit Hilfe dieses Standards sollten nun Genauigkeiten im Sub-Mikrosekunden Bereich möglich werden. 2008 wurde dann der Standard nochmals an die aktuellen Anforderungen angepasst und als IEEE 1588-2008 festgeschrieben. Dieser Standard ist bis heute gültig. Zur Zeit arbeitet die IEEE Working Group um John C. Eidson und Doug Arnold, der nun als Co-Chairmen fungiert, an einer neueren Version des Standards um die erweiterten Anforderungen an den Standard Rechnung tragen zu können. Folgende neue Anforderungen wurden für den neuen Standard identifiziert:

- Verbesserung der Zeitsynchronisation Genauigkeit
- Anwendbarkeit auf lokalisierte Systeme mit der Möglichkeit der Verwendung in größeren Systemen
- Administrationsfreien Betrieb
- Eignung in High-End Geräten sowie in Low-End Geräten
- Management-Steuerung für redundante und fehlertolerante Systeme

Doch nicht nur in der IEEE Working Group wurde an der Verbesserung des Standards gearbeitet. Ein Zusammenschluss verschiedener Institutionen und Unternehmen, darunter die CERN-Forschungsorganisation, das GSI Helmholtz-Zentrum für Schwerionen Forschung und die Firma National Instruments, haben den Standard für ihre Anforderungen weiterentwickelt und unter dem Projektnamen „White Rabbit“ [3] veröffentlicht. Die Ziele des White Rabbit-Projektes waren [4]:

- Genauigkeit der Zeitsynchronisation im Sub-Nanosekundenbereich
- Tausende von Knoten innerhalb der PTP Domäne
- mögliche Knotenabstände von mehr als 10 km
- Ethernet-basierte Gigabit-Netzwerk mit zuverlässige Datenübertragung
- Open-Source-Hardware, Firmware und Software
- Handelsübliche Hardware verschiedener Hersteller

Die aktuelle Version 2.0 des White Rabbit Standards wurde 2011 unter der "CERN Open Hardware License" veröffentlicht. Derzeit wird dieser Standard bereits in verschiedenen Projekten eingesetzt. CERN verwendet den Standard für ein Steuerungssystem des "Large Hadron Collider" und für sein Neutrino - Teleskop KM3Net im Mittelmeer und das "GSI Helmholtzzentrum für Schwerionenforschung" nutzt es für den Partikel Beschleuniger FAIR.

Um den IEEE 1588 Standard besser verstehen zu können, sollte man sich nochmal die Probleme vor Augen führen, wieso eine derartig genaue Zeitsynchronisation so komplex sein kann. Im Grunde bestehen mehrere Probleme:

- 1) Jede Uhr innerhalb eines Computers ist als Zähler implementiert. Es werden die Takte einer Taktquelle gezählt und über die Frequenz der Taktquelle die Uhrzeit errechnet. Da Taktquellen nie einen konstanten Takt erzeugen können, können auch zwei parallel laufende Taktquellen nie den identischen Takt haben. Somit ist eine Anpassung der Taktfrequenz des Slaves zum Master notwendig.
- 2) Das im Regelfall verwendete Ethernet entspricht weitestgehend der IEEE 802.3. Ethernet kann leider keine zeitlich deterministische Verbindung zwischen zwei Rechnern bereitstellen. Die Laufzeiten können von Sendung zu Sendung variieren. Somit können auch die Laufzeiten der Synchronisationsdaten variieren, was zu Fehlern bei der Synchronisation führen kann.
- 3) Die Laufzeiten innerhalb der Rechner, also in den Treibern und dem Network-Stacks der Betriebssysteme, sind ebenfalls nicht deterministisch. Moderne Betriebssysteme arbeiten mit „preemptive scheduling“, was heißt, dass jeder Prozess innerhalb des Betriebssystems unterbrochen werden kann und zu einer anderen Zeit fortgesetzt wird. Somit können Verarbeitungszeiten innerhalb des Betriebssystems nicht mehr deterministisch vorhergesagt werden. Diese Unschärfe führt wiederum zu Fehlern in der Zeitsynchronisation und muss berücksichtigt werden.

- 4) Als Übertragungsmedien muss ein Consumer Produkt (Netzwerkkarte, Netzwerk-Chip) verwendet werden. Es müssen die bestehenden Infrastrukturen (Netzwerke) verwendet werden. Es darf also keine eigenständige Hardware für die Zeitsynchronisation verwendet werden

Der IEEE-1588-2008 Standard enthält verschiedene Aspekte der Synchronisation und ihrer Umgebung. Zuerst werden die Geräte in den Netzwerken betrachtet (PTP-Domänen) und deren Anordnung. Jedes der Geräte oder Knoten in einer PTP-Domäne die eine PTP Funktionalität enthalten, werden als „Clocks“ bezeichnet und können in vier verschiedenen Varianten auftreten.

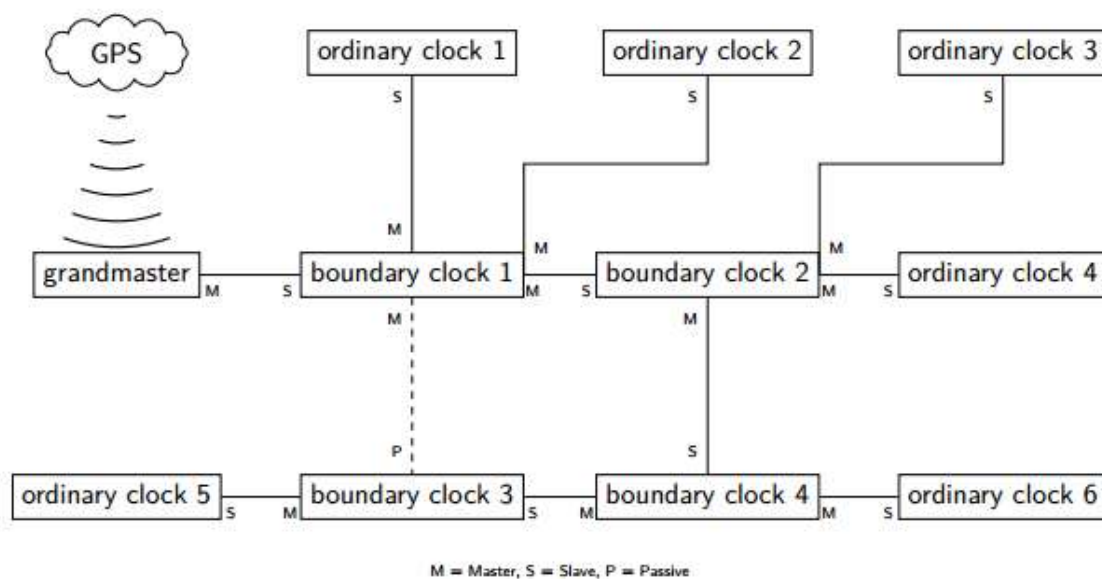


Bild 1: Schematischer Aufbau eines PTP Netzwerkes [2]

### Ordinary Clock:

Eine „Ordinary Clock“ ist eine einfache Uhr (ein Port, meist ein Client) der als Slave mit einem Master verbunden ist und ihre Zeit an dessen angleicht.

### Boundary Clock:

Eine „Boundary Clock“ ist eine Uhr die mehrere „Ordinary Clocks“ enthält (mehrere Ports) und die als Master mehrere Slaves mit ihrer Zeit synchronisieren kann. Die „Boundary Clock“ kann aber auch als Slave mit einem Master verbunden sein und ihre Zeit an dessen angleichen.

### Transparent Clock:

Eine „Transparent Clock“ ist eine Uhr die nicht aktiv in die Zeitsynchronisation eingreift, sie ist mehr eine Hardware die die Zeitsynchronisation-Datenpakete vermittelt, typischerweise ein Netzwerk Switch. „Transparent Clocks“ korrigieren die Zeitstempel innerhalb der Datenpakete um die Verweildauer innerhalb der Hardware.

### **Grandmaster Clock:**

Die Grandmaster Clock ist eine „Ordinary Clock“, die normalerweise Zugang zu GPS oder einer anderen sehr genauen Zeit hat und diese sehr genaue Zeit für alle untergeordneten Knoten bereitstellt.

Innerhalb des PTP Netzwerkes kann jede Clock (Port) entweder als Master oder Slave agieren. Innerhalb einer PTP Domäne kann es aber immer nur einen Master geben. Der Master verteilt aktiv seine Zeit an die Slaves. Er ist somit für die Synchronisation der Slaves in seiner PTP Domäne zuständig. Ein Slave ist ein passives Element innerhalb von PTP der auf die Zeitsynchronisation-Anfragen des Masters antwortet. Grundsätzlich kann jede Clock (Port) innerhalb von PTP als Master oder Slave arbeiten. Die Entscheidung ob die Clock als Master oder Slave arbeiten muss wird beim Betreten des PTP Netzwerkes ermittelt. Hierzu wurde der „Best Master Clock“-Algorithmus in IEEE 1588 Standard definiert. Dieser Algorithmus regelt, dass die Clock mit der besten Genauigkeit, die Rolle des Masters übernimmt.

Die Synchronisation der Uhrzeit zwischen Master und Slave wird über sogenannte „Event Messages“ durchgeführt. Diese Messages werden über das UDP Netzwerkprotokoll im Netzwerk vom Master verteilt und von Slaves beantwortet. Es existieren zwei verschiedene Verfahren für die Synchronisierung, die „One-Step“ Synchronisierung und die „Two-Step“ Methode. Der Unterschied in den beiden Methoden liegt in der Anzahl der Nachrichten welche ausgetauscht werden müssen und ist im kommenden Diagramm zu sehen. Die „Two-Step“ Methode verschickt vier Nachrichten um einen Synchronisationsschritt durchzuführen. Dahingegen braucht die „One-Step“ Methode nur drei Nachrichten, was aber nur mit äußerst guter Hardware-Unterstützung möglich ist und somit nicht in jeder PTP Domäne gegeben sein kann.

Damit ein Slave die Synchronisierung durchführen kann, sind lediglich vier Zeitstempel ( $t_1, t_2, t_3, t_4$ ) nötig. Die Zeitstempel werden jeweils beim Senden oder Empfangen bestimmter Nachrichten generiert. Der Slave erhält diese vier Zeitstempel durch den Austausch der folgenden „Messages“:

- Sync Message, der Master ermöglicht damit den Slaves, eine Synchronisierung zu beginnen; In der eben genannten „One-Step“ Methode, ist in dieser Nachricht der Zeitstempel  $t_1$  enthalten
- Follow\_Up Message, diese Message ist analog zur Sync Message zu sehen, wobei damit nun der Zeitstempel  $t_1$  verschickt wird, falls die „Two-Step“ Methoden angewendet wird
- Delay\_Req Message, der Slave antwortet hiermit auf eine Sync oder Follow\_Up Nachricht und erfragt damit den Master um einen weiteren Timestamp
  - Delay\_Resp Message, der Master schickt mit dieser Message den letzten Zeitstempel  $t_4$  an den Slave zurück

Das nachfolgende Diagramm zeigt den genauen Ablauf der PTP Kommunikation und den damit verbundenen Austausch der benötigten Zeitstempel. Die Erfassung der Zeitstempel  $t_1$  und  $t_4$  geschieht auf der Seite des Masters und wird mit den Nachrichten an die Slaves gesendet. Damit dieser Knoten zum Schluss seine vier Zeitstempel besitzt, werden  $t_2$  und  $t_3$  Slave-seitig bei Empfang der Master Nachrichten erstellt.

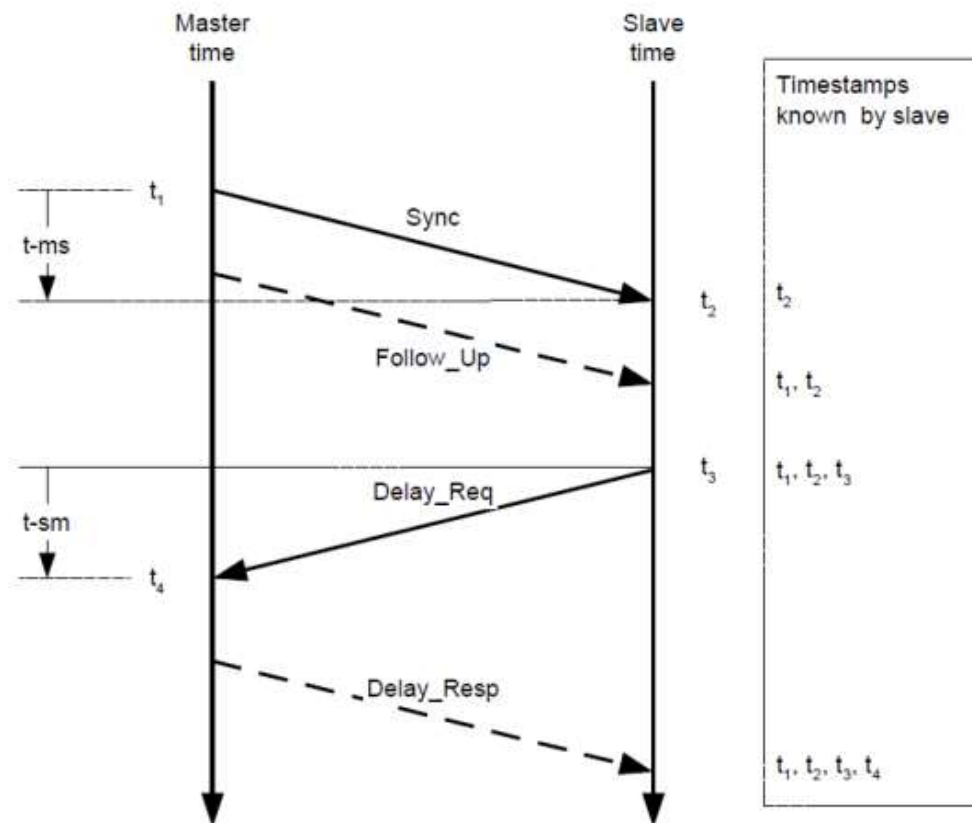


Bild 2: Schematischer Ablauf der Messages[1]

Mit Hilfe der Zeiten  $t_1$ ,  $t_2$ ,  $t_3$ , und  $t_4$  kann nach dem Austausch der Nachrichten der „meanPathDelay“ berechnet werden. Der „meanPathDelay“ gibt an, wie lange eine Message für die Übertragung im Netzwerk gebraucht hat, also die Verzögerung. Wie an der untenstehenden Formel zu sehen ist, geht IEEE 1588 davon aus das die Latenzzeit im Netzwerk für Senden und Empfangen identisch ist.

$$\langle \text{meanPathDelay} \rangle = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} = \frac{t_{ms} + t_{sm}}{2}$$

Bild 3: Formel zu Berechnung des meanPathDelay [2]

Somit dient der „meanPathDelay“ zur Laufzeitkorrektur (im Netzwerk) für die Berechnung des eigenen Zeit-Offsets.

Wenn man das obige Diagramm betrachtet kann man zum Schluss kommen, dass der berechnete „meanPathDelay“ umso genauer ist, je näher der Zeitstempel  $t_n$  an der Hardwareschnittstelle gesetzt oder ermittelt wird. Diese Beobachtung führt dann sehr schnell zur Erkenntnis, dass ein Teil der IEEE 1588 Funktionalität optimaler Weise in den entsprechenden Netzwerkchips untergebracht werden sollte. 2012 hat Intel dann ihre Netzwerkchips der Intel Reihe I21x vorgestellt die erstmals eine Unterstützung für IEEE 1588 enthielten. Mit der Vorstellung dieser Netzwerkchips war nun auch der Zeitpunkt gekommen um über kommerzielle Lösungen im T&M Bereich nachzudenken. Intel hatte mit diesem Consumer Chipsatz erstmals die Möglichkeit geschaffen für eine breite Palette an Geräten die Verwendung von IEEE 1588 zu ermöglichen. Diese Netzwerkchips enthalten alle zeitkritischen Teile des IEEE 1588 Standards:

- Hardware Clock: Einen internen Zähler und die dazu notwendigen Stellglieder für die Anpassungen des Zählers (Frequenzanpassung)
- Hardware Timestamping: Das Timestamping der zusendenden UDP PTP Datenpakete
- Hardware Timestamping Grabber: Das Erfassen der Timestamps beim Empfangen von UDP PTP Datenpakete
- Hardware Based Time Events: Das Erzeugen von zeitgesteuerten Trigger

Unter Linux existieren bereits einige Implementierungen von unterschiedlicher Qualität. Für das Betriebssystem Windows existieren jedoch keinerlei Treiber von Microsoft oder Intel um diese Funktionalität der Intel I21x Netzwerkchips zu unterstützen. Zwar sind einige käufliche Implementierungen erhältlich, die preislich jedoch nur eingeschränkt attraktiv sind.

Mit Hilfe der Intel I21x Netzwerkchips lassen sich mit geringem Aufwand schon hervorragende Zeitsynchronisationen in zweistelligen Nanosekunden Bereich erreichen, die auch langzeitstabil gehalten werden können.

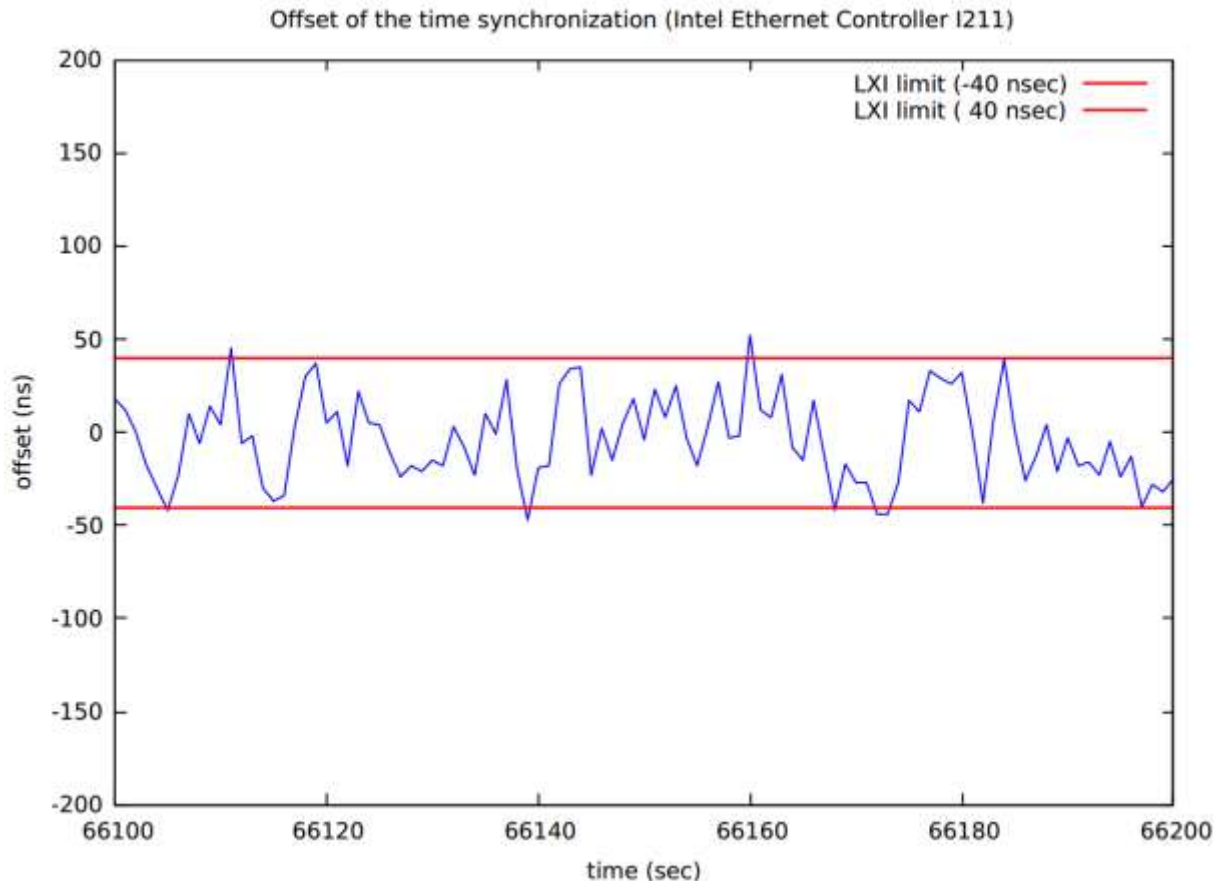


Bild 4: Messung des Slave Offsets über 100 Sekunden [2]

Wie das Diagramm oben zeigt schwankt der Offset zwischen Master und Slave immer um den Nullpunkt. Aufgrund des berechneten Offsets muss die aktuelle Uhrzeit des Slaves angepasst werden. Wie leicht einzusehen ist, kann nicht einfach die Uhrzeit korrigiert werden, also der ermittelte Offset direkt angewandt werden, da die Zeit dann in die Vergangenheit oder Zukunft springen würde. Ein sinnvollerer Vorgehen ist hier, den Takt des Zählers im Slave zu manipulieren. Die Intel I21x Chips haben hierzu ein spezielles Register mit dem die Frequenz des Zählers indirekt angepasst werden kann. Der Intel I21x Chip aktualisiert alle 8 ns seine lokale Uhr um 8ns, zusätzlich kann ein Offset von  $n \times 2^{-31}$  ns definiert werden, die bei jedem dieser Updates hinzuaddiert werden. Dieser Offset ist vorzeichenbehaftet und kann somit zu Verkleinerung oder Vergrößerung der Zählerfrequenz verwendet werden. Aufgrund des Offsets zum Master kann nun der entsprechende Fehler in der Frequenz und die somit notwendige Frequenzanpassung ermittelt werden. Der zugrundeliegende Algorithmus der Frequenzanpassung ist maßgeblich für die Stabilität der Uhr verantwortlich.

Innerhalb des Netzwerkes kann es auch zu Störungen kommen die letztlich die Stabilität der Clock beeinflussen. Das unten aufgeführte Diagramm zeigt Störungen die von Netzwerkkollisionen und die sich dadurch ergebenden Übertragungswiederholungen im Ethernet entstanden sind.



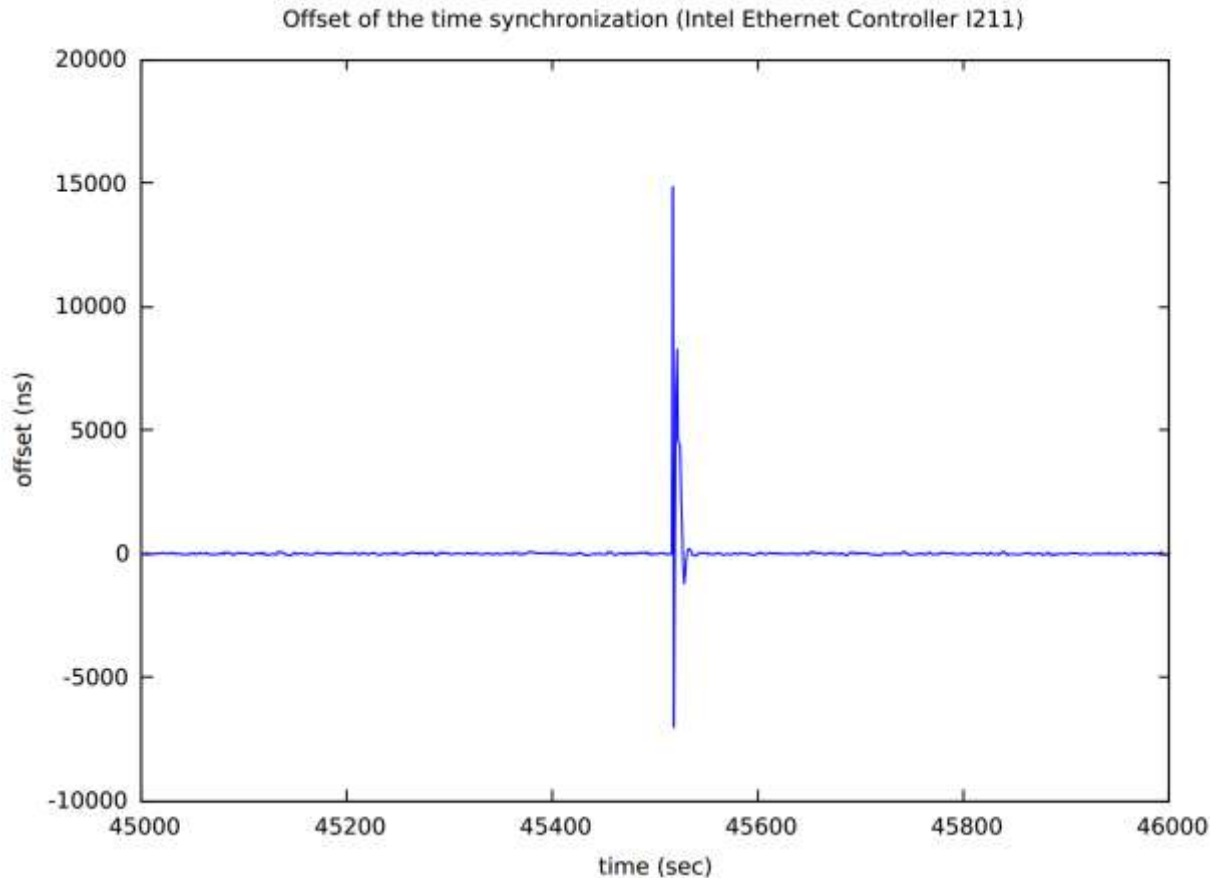


Bild 5: Messung des Slave Offsets über 100 Sekunden [2]

Um derartige Störgrößen erkennen und bewerten zu können, kann zum Beispiel ein Kalman-Filter eingesetzt werden.

Im T&M Bereich werden heute noch viele Messaufgaben mit Hilfe des klassischen Vorgehens gelöst. Mit Hilfe der genauen Zeitsynchronisation über IEEE 1588 ergeben sich nun aber viele neue und interessante Lösungsansätze.

Wenn man sich vor Augen hält, dass ein Triggersignal ca. 5ns Laufzeit pro Meter benötigt, kann man sich sicherlich einige Messaufgaben vorstellen die mit Hilfe der IEEE 1588 Zeitsynchronisation besser und genauer gelöst werden können. Bei Messaufgaben die beispielsweise mit Satellitensignalen zu tun haben, kann es durchaus vorkommen dass Messgeräte 100 und mehr Meter voneinander entfernt sind, hier wäre der Ansatz mit IEEE 1588 sicherlich eine Alternative.

Auch im Zuge der zunehmenden Verbreitung der IoT Sensoren und der Offline Messdatenaufnahme, wäre eine genaue und stabile Zeitsynchronisation der einzelnen Messsensoren von großem Vorteil. Somit können die einzelnen IoT Sensoren kontinuierlich ihre Daten liefern und zum Beispiel in einer Cloud ablegen. Diese Daten können dann bei einer Offline Messdatenauswertung über den Zeitstempel, der bei der Messungen erstellt worden ist, einfach korreliert werden.

Da IEEE 1588 auch zeitgesteuerte Events definiert, können zum Beispiel mehrere Geräte miteinander Sequenzen von Messaufgaben abarbeiten ohne umständlich mit Triggerleitungen oder ähnlichem verbunden sein zu müssen. So könnte zum Beispiel ein Signalgenerator seine Frequenzliste zeitgesteuert abarbeiten und der zugehörige Analysator seine Messungen durchführen. Beide Aufgaben würden dann durch die zeitgebundenen Events gesteuert werden.

Auch im Bereich Mobilfunk kann man mit dem Einsatz von IEEE 1588 sicherlich verschiedenste Messaufgaben deutlich einfacher realisiert werden, da die zeitliche Kopplung zwischen Basisstation und Mobilfunkgerät bei diversen Messaufgaben von Relevanz ist.

Diese Anwendungsbeispiele hören sich im ersten Ansatz alle sehr vielversprechend an, basieren jedoch auf der Annahme dass auch die Messgeräte-Hersteller diesen Standard in ihren Geräten implementiert haben. Selbst wenn alle den IEEE 1588 Standard verwenden würden, ist immer noch eine weitergehende Standardisierung notwendig um die einzelnen Aktionen und Aufgaben zu koordinieren.

Um dies den Messgeräte-Herstellern zu erleichtern wurde 2004 der LXI Standard (Lan eXtension for Instruments) von dem führenden Messgeräte-Hersteller ins Leben gerufen. 2008 wurde dann im LXI Standard der IEEE 1588 Standard mit aufgenommen und zusätzlich um einige Erweiterungen, wie die Lan Event Messages, erweitert. Mit Hilfe dieses Standards ist es nun den Messgeräte-Herstellern möglich die Vorteile von IEEE 1588 in Messgeräten effizient einzusetzen. Bis zur Vorstellung der Intel I21x Netzwerkchips war immer kostenintensive oder eigenentwickelte Hardware notwendig um den IEEE 1588 Standard in den Messgeräten zu integrieren. Mit den Intel Netzwerkchips bietet sich nun erstmals die Möglichkeit diesen Standard mit Consumer Elektronik zu realisieren.

Bereits 2014 entschied das LXI Konsortium für Ihre Mitglieder ein Referenz Design bereitzustellen, um die Verwendung des Standards weiter zu verbreiten. Nachdem im LXI Referenz Design 2016 die Core Funktionalität des Standards verfügbar waren, begann das LXI Konsortium mit der deutschen Firma TSEP die Möglichkeit einer Referenz Implementierung der LXI Time Synchronisation (entspricht IEEE 1588) zu evaluieren und einen Prototypen zu erstellen. Im Februar 2017 wurde dann von TSEP auf einem der regelmäßigen Treffen des LXI Konsortium in den USA der erste Prototyp vorgestellt. Sowohl unter Linux als auch unter Windows war der Prototyp auf einem käuflichen Rechnerboard mit Intel I211 Chip lauffähig.

Für den Linux Prototyp wurde das Linux Packet „LinuxPTP“ verwendet. Die Implementierung der eigentlichen Clock Funktionalität war aber nicht ausreichend um die geforderten Rahmenbedingungen des LXI und IEEE 1588 Standards zu erfüllen. TSEP hat deshalb dieses nachimplementiert. Der Regelalgorithmus zur

Frequenznachführung der IEEE 1588 Clock wurde dabei komplett neu entwickelt und optimiert.

Wie bereits erwähnt, bietet Windows keinerlei Unterstützung für IEEE 1588, deshalb wurde sowohl die komplette Clock Funktionalität als auch der Treibersupport von TSEP erstellt. TSEP hat hierfür die notwendigen Treiber für die Ansteuerung der Intel I21x Netzwerkchips für das Timestamping und das Erkennen von IEEE 1588 Paketen im Windows Network Stack verankert. Sowohl unter Windows 7 als auch unter Windows 10 konnte diese Funktionalität getestet werden.

Die ersten Gespräche mit dem LXI Konsortium über die Bereitstellung der LXI Time Synchronisation (entspricht IEEE 1588) waren durchweg positiv und werden wahrscheinlich zu einer kostenlosen Bereitstellung für LXI Mitglieder in diesem Jahr führen. TSEP plant zusätzlich seine Implementierung für andere Interessenten, die nicht LXI Mitglieder sind, bereitzustellen.

Zusammenfassend kann gesagt werden, dass mit dem kommerziellen Einsatz der Intel I21x Netzwerk-Chips es nun möglich ist IEEE 1588 in T&M Geräten einzusetzen. Die neuen Lösungsansätze die sich mit IEEE 1588 ergeben, erweitern die Möglichkeit Messaufgaben im T&M Bereich zu lösen um eine neue Dimension. Wird zusätzlich noch Erweiterungen wie der LXI Standard in T&M Geräten angeboten, lassen sich komplexe Messaufgaben einfacher und effizienter strukturieren. Selbstkonfigurierende Systeme sind mit diesen Ansätzen nun möglich, da Geräte Daten und Aktionen Netzwerkweit konfigurieren, austauschen und auslösen können. Es bleibt abzuwarten wie die Messgeräte-Hersteller auf diese neue Technologie reagieren, ein großes Potenzial steckt allemal in ihr.

### **Literaturverzeichnis:**

[1] IEEE. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems, 2008.

[2] Bachelor Arbeit von Dominik Richstein, „Usability of the Intel Ethernet Controllers I210, I211 and I350 for the LXI Extended Function Clock Synchronization“

[3] CERN. White rabbit specification: Draft for comments, 06.07.2011.

[4] White Rabbit Project. White rabbit wiki, 2016.

### **Autoren:**

#### **Peter Plazotta, Dipl. Ing (FH)**

CEO von TSEP und seit 30 Jahren mit dem Design und der Entwicklung von Systemsoftware vor allen im den Bereichen T&M, Telekommunikation und Automotive beschäftigt.

Peter.Plazotta@tsep.com

#### **Dominik Richstein, B. Sc,**

Studierte an der Technischen Universität von Ingolstadt und hat seine Bachelorarbeit über das Thema IEEE 1588 und Intel I21x Netzerkchips bei TSEP geschrieben.

Dominik.Richstein@tsep.com